International Journal of Technological Exploration and Learning (IJTEL)
Volume 2 Issue 3 (June 2013)

Distributed Cut Detection in Wireless Sensor Networks

K. R. Lokesh
Department of Computer Science and Engineering,
Intell Engineering College,
Anantapur, Andhra Pradesh,India

C.Nagesh
Associate Professor,
Department of CSE,
Intell Engineering College,
Anantapur, Andhra Pradesh,India

P.Namratha
Assistant Professor,
Department of CSE,
Intell Engineering College,
Anantapur, Andhra Pradesh,India

Abstract— **A wireless sensor network consists of spatially distributed autonomous sensors to physical or environmental conditions, such as temperature, sound, pressure, etc. and to cooperatively pass their data through the network to main location. When the data transmission between one node to another, there is some of the nodes in the network are failed, because of above mentioned issues. So, the data will not be transferred to the destination. Because of these failures of nodes the network is divided into multiple parts. The ability to detect the cuts by using the source node and disconnected node of a wireless sensor network will lead to the increase in the lifetime of network. The Distributed Cut Detection (DCD) algorithm proposed here enables every node of wireless sensor network to detect Disconnected from Source events if they occur. Second, it enables a subset of nodes that experience CCOS events to detect them and find the approximate location of the cut in the form of a list of active nodes that lie at the boundary of the cut. The advantage of DCD algorithm is that convergence rate of the iterative scheme is quite fast and independent of the size and structure of network.**

Keywords- Wireless Sensor Networks; Cut in Wireless Networks; Ddetection and Estimation; Iterative computation.

I. INTRODUCTION

Wireless sensor networks (WSNs) are a promising technology for monitoring large regions at high spatial and temporal resolution. However, the small size and low cost of the nodes that makes them attractive for widespread deployment also causes the disadvantage of low-operational reliability. The node may fail due to different problems. In fact, node failure is expected to be quite common due to the typically limited energy budget of the nodes that are powered by small batteries. Failure of a set of nodes will reduce the number of multi hop paths in the network. Such failures cause a subset of nodes—that have not failed—to become disconnected from the rest, resulting in a ”cut”.

We consider the problem of detecting cuts by the nodes of a wireless sensor network. May source node is a base station serves as an interface between the network and its users. So, cut may or may not separate a node from the source node, when a node is disconnected from the source is u, when a cut occurs in the network that does not separate a node u from the source node, we say that these nodes are connected, but a cut occurred somewhere (CCOS) event has occurred for u. By cut detection we mean 1) detection by each node of DOS event when it occurs, and 2) detection of CCOS events by the nodes close to a cut, and the approximate location of the cut. Nodes that detect the occurrence and approximation locations of the cuts can then alert the source node or the base station. if a node having the ability to detect the cut, it could simply wait for the network to be repaired and eventually reconnected, so it saves the energy of the multiple nodes after cut.

In this paper we propose a distributed algorithm to detect cuts, named the Distributed Cut Detection (DCD) algorithm. The algorithm allows each node to detect DOS events and a subset of nodes to detect CCOS events. The algorithm we propose is distributed and asynchronous: it involves only local communication between nodes, and is robust to temporary communication failure between node pairs. A key component of the DCD algorithm is a distributed iterative computational step through which nodes compute their electrical potentials. The convergence rate of the computation is independent of the size and structure of the network.

II. PROBLEM STATEMENT

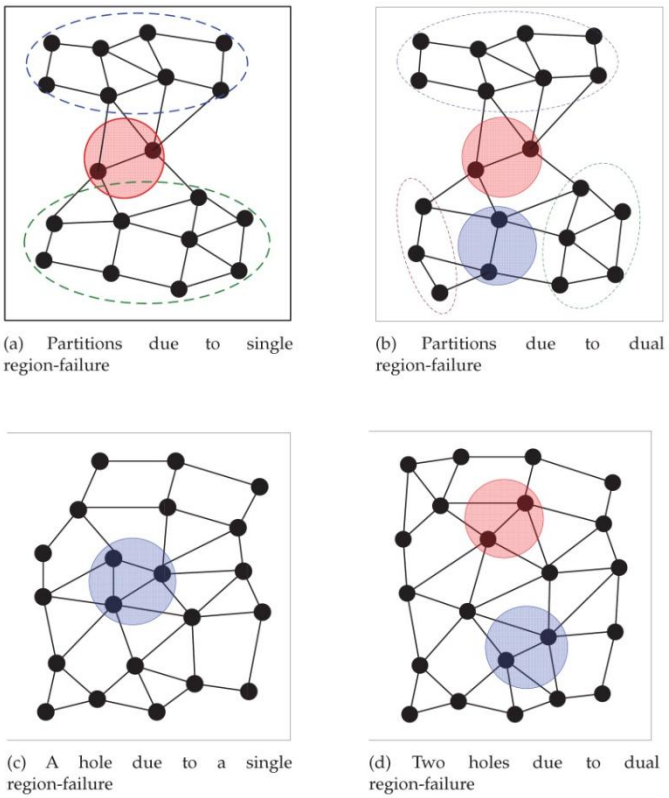
Consider a sensor network modeled as a undirected graph G=(V,E), whose node set V represents the sensor nodes and the edge set E consists of pair of nodes (u, v) such that nodes u and v can exchange messages between each other. Note that we assume inter-node communication is symmetric.

An edge (u, v) is said to be incident on both the u and v. The nodes that share an edge with a particular node u are called the neighbors of u. A cut is the failure of a set of nodes V cut from G results in G being divided into multiple connected components. Recall that an undirected graph is said to be connected if there is a way to go from every node to every other node by traversing the edges and that a component GC of a graph G is a maximal connected sub graph of G. We are interested in devising a way to detect if a subset of the nodes has been disconnected from a distinguished node, which we call the source node, due to the occurrence of a cut.

III. DISTRIBUTED CUT DETECTION

The algorithm is based on an electrical analogy. Given an undirected graph G = (V , E) with, say, n nodes m edges that

describes the sensor network, this algorithm is used for the nodes which is disconnected from the source node. We construct the graph Gelec = (Velec , E elec) where Velec =V U Vfict, where Vfict consists of n - 1 nodes, one node for every node in V except the source node, and V is connected to it fictitious node in Vfict with a single edge.



(a) Partitions due to single region-failure

(b) Partitions due to dual region-failure

(c) A hole due to a single region-failure

(d) Two holes due to dual region-failure

Figure 1. A graph describing a sensor network and associated electrical network.

In this algorithm we are having two phases. One is state update law, it works very efficient to calculate the node potentials in electrical network (Gelec,1) when s Ampere current is injected to the source node and extracted to the nodes Vfict, with all nodes in V . The other phase of the algorithm consists of monitoring the state of a node, it is used to detect the cut occurred. Now we describe about the each phase.

A. State update law

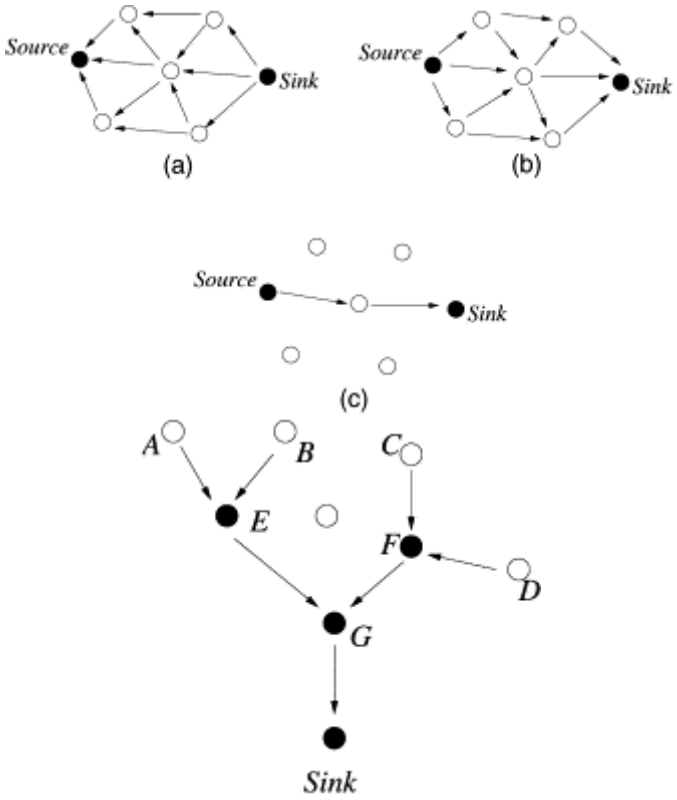
The nodes use the computed potentials to detect if DOS events have occurred (ie., if they are disconnected from the source node). To detect CCOS events, the algorithm uses the fact that the potentials of the nodes that are connected to the source node also change after the cut. CCOS detection proceeds by using probe messages that are initiated by certain nodes that encounter failed neighbors, if a short path exists around a "hole" created by node failures, the message will reach the initiating node.

Every node keeps a scalar variable, which is called a state Let G(k) = (V(k),E(k)) represent the sensor network that consists of all the nodes and edges of G that are still active at time k, where k=| 0,1,2,…. is an iteration counter. We index the source node as 1. Every node u maintains a scalar state xu(k) that is updated. At every iteration k, nodes broadcast their current states. Let Nu(k) = {v|(u,v) € E(k)} denote the set of neighbors of u in the graph G(k). Every node in V except the source node updates the following state law. Where strength is design parameter:

$$Xi(k+1) = \frac{1}{Di(k)+1} \sum_{j \in Ni(k)} xj(k) + s1\{1\}(i), \quad (1)$$

Where di(k) := | Ni(k) | is the degree of node i at time k, and 1A(i) is the indicator function of the set A. That is, 1{1} (i) = 1 if i=1, and 1{1}(i) = 0 if i=1 and. After that, i can update its neighbor list Ni(k) as follows: if no messages have been received from a neighboring node for the past Tdrop iterations, node i drops that node from list of neighbors. The integer parameter Tdrop is a design choice.

When the network is connected, the state of a node converges to its potential in the electrical network (Gelec ,1), which is a positive number. The potential of a node that is disconnected from the source is 0; this is the value converges to . If the reconnection occurs after a cut, the states of reconnected nodes again converge to positive values. Especially with wireless communication an asynchronous update is preferable



The above graph represents the sensor network and the associated sensor network.

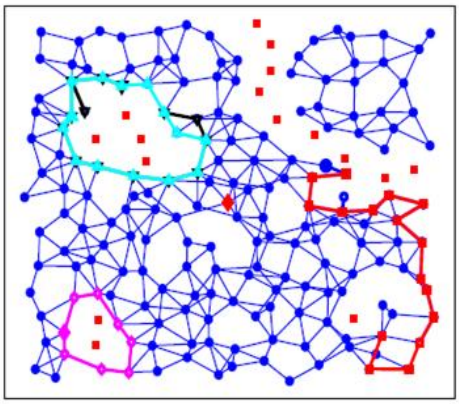
B. State monitoring for cut detection

Theorem 1 shows how the occurrence of a cut in the Network is manifested in the states of the nodes. By analyzing their own states, nodes can detect if a cut has occurred.

Suppose a cut occurs at some time $\tau > 0$ which separates the network into n components Gsource, G2, . . . , Gn, the component G source containing the source node. Since there is no source (and therefore no current injection) in each of the components G2, . . . , Gn disconnected from the source, it follows from Theorem 1 that the state of every node in each of these components will converge to zero. When the potential at a particular node drops below a particular threshold value, the node can declare itself cut from the source node. In fact, there

may be additional node failures (and even increase in the number of components) after the cut appears. Since the state of a node converges to 0 if there is no path to the source, additional time variation in the network will not affect cut detection. If additional failures do not occur after the cut occurs, it follows from Theorem 1 that the states of the nodes that are in the component G source (which contains the source) will converge to new steady state values. So, if a node detects that its state has converged to a steady state, then changed, and then again converged to a new steady state value that is different from the initially seen steady state, it concludes that there has been a cut somewhere in the network.A node detects when steady state is reached by comparing the derivative of its state (with respect to time) with a small number ϱ that is provided a-priori. The parameters s and ϱ are design variables. It updates its state from that neighbor, in the asynchronous setting every node keeps a local iteration counter that may differ from those of other nodes by arbitrary amount.

The source node is at the center. The nodes 3b fail at k = 100, and thereafter they do not participate in communication or computation. Figs. The state of a node u decays to 0 after reaching a positive value, where the state of node v says positive.



A) G(k) for k>100

IV. DISTRIBUTED CUT DETECTION ALGORITHM

A. *Dos Detection*

We say that a Disconnected from Source (DOS) event has occurred for u. The algorithm allows each node to detect DOS events. The nodes use the computed potentials to detect if DOS events have occurred (i.e., if they are disconnected from the source node). The approach here is to exploit the fact that if the state is close to 0 then the node is disconnected from the source, otherwise not. In order to reduce sensitivity of the algorithm to variations in network size and structure, we use a normalized state.DOS detection part consists of steady-state detection, normalized state computation, and connection/separation detection. A node keeps track of the positive steady states seen in the past using the following method. Each node computes the normalized state difference () as follows:

$$xi(k) = \begin{cases} \dfrac{xi(k) - xi(k-1)}{Xi(k-1)}, & \text{if } xi(k-1) > \epsilon \text{ zero} \\ \infty, & \text{otherwise,} \end{cases}$$

Where, zero is a small positive number.

A node keeps a Boolean variable Positive Steady State Reached (PSSR) and updates PSSR(k)← 1 if /( )| < for = k − Tguard , k −Tguard +1,…,k(i.e., for Tguard consecutive iterations),where is a small positive number and Tguard is a Small integer. The initial 0 value of the state is not considered a steady state, so PSSR()=0 for =0,1, … ,Tguard. Each node keeps an estimate of the most recent "steady state" observed, which is denoted by ( ). This estimate is updated at every time k according to the following rule: if PSSR( )=1, then ← ; otherwise ← − 1 . It is initialized as ss(0) = ∞.Every node i also keeps a list of steady states seen in the past, one value for each unpunctuated interval of time during which the state was detected to be steady. This information is kept in a vector ( ), which is initialized to be empty and is updated as follows: If PSSR () = 1 but PSSR(−1) = 0, then is appended to ( ) as a new entry. If steady state reached was detected in both and - 1 (i.e., PSSR( ) = PSSR(− 1) = 1, then the last entry of ( ) is updated to ( ) .

B. *CCOS Detection*

When a cut occurs in the network that does not separate a node u from the source node, we say that Connected, but a Cut Occurred Somewhere (CCOS) event has occurred for u. detection of CCOS events by the nodes close to a cut, and the approximate location of the cut. By "approximate location" of a cut we mean the location of one or more active nodes that lie at the boundary of the cut and that are connected to the source. To detect CCOS events, the algorithm uses the fact that the potentials of the nodes that are connected to the source node also change after the cut. However, a change in a node's potential is not enough to detect CCOS events, since failure of nodes that do not cause a cut also leads to changes in the potentials of their neighbors. Therefore, CCOS detection proceeds by using probe messages.

V. SYSTEM IMPLEMENTATION

In this section, we describe the software implementation and evaluation of the DCD algorithm. In software the algorithm was implemented using the java language running on windows xp operating system. The system executes in two phases: the Reliable Neighbor Discovery (RND) phase and the DCD Algorithm phase. In the RND phase each node is connected to the source node. Upon receiving the message, the mote updates the number of beacons received from that particular sender. To determine whether a communication link is established, each mote first computes for each of its neighbors the Packet Reception Ratio (PRR), defined as the ratio of the number of successfully received beacons and the total number of beacons sent by a neighbor. A neighbor is deemed reliable if the PRR >0:8. Next, the DCD algorithm executes. After receiving state information from neighbors, a node updates its state according to (1) in an asynchronous manner and broadcasts its new state. The state is stored in the database.

CONCLUSION

The DCD algorithm we propose here enables every node of a wireless sensor network to detect Disconnected from Source events if they occur. Second, it enables a subset of nodes that experience CCOS events to detect them and estimate the approximate location of the cut in the form of a list of active nodes that lie at the boundary of the cut/hole.The DOS and CCOS events are defined with respect to a specially designated source node. The algorithm is based on ideas from electrical network theory and parallel iterative solution of linear

equations. The algorithm works effectively with large classes of graphs of varying size and structure, without requiring changes in the parameters. For certain scenarios, the algorithm is assured to detect connection and disconnection to the source node without error. A key strength of the DCD algorithm is that the convergence rate of the underlying iterative scheme is quite fast and independent of the size and structure of the network, which makes detection using this algorithm quite fast. Publication of the DCD algorithm to detect node separation and reconnection to the source in mobile networks is a topic of ongoing research.

REFERENCES

[1] G.Dini, M. Pelagatti, and I.M. Savino, "An Algorithm for Reconnecting Wireless Sensor Network Partitions," Proc. European Conf. Wireless Sensor Networks, pp. 253-267, 2008.

[2] N. Shrivastava, S. Suri, and C. D. T´oth, "Detecting cuts in sensor networks,"in IPSN '05: Proceedings of the 4th international symposiumon Information processing in sensor networks, 2005, pp. 210–217.

[3] Ö. Babaoglu, R. Davoli, A. Montresor, Group Communication in Partitionable Systems: Specification and Algorithms, IEEE Transactions on Software Engineering,.

[4] H. Ritter, R. Winter, and J. Schiller, "A Partition Detection System for Mobile Ad-hoc Networks," Proc. First Ann. IEEE Comm. Soc. Conf. Sensor and Ad Hoc Comm.and Networks (IEEE SECON '04), pp. 489-497, Oct. 2004.

[5] M. Hauspie, J. Carle, and D. Simplot, "Partition Detection in Mobile Ad-Hoc Networks," Proc. Second Mediterranean Workshop Ad-Hoc Networks, pp. 25-27,

[6] P. Barooah, "Distributed Cut Detection in Sensor Networks," Proc. 47th IEEE Conf. Decision and Control,pp. 1097-1102, Dec.2008.

[7] A.D. Wood, J.A. Stankovic, and S.H. Son, "Jam: A Jammed-Area Mapping Service for Sensor Networks," Proc. IEEE Real Time Systems Symp., 2003. International Journal of Engineering Research & Technology (IJERT) Vol. 1 Issue 10, December- 2012 ISSN: 2278-0181.

[8] G. H. Golub and C. F. van Loan, Matrix Computations, 3rd ed. The John Hopkins University Press, 1996.

[9] F. Chung, "Spectral graph theory," Regional Conference Series in Mathematics, Providence, R.I., 1997.

[10] A. Jadbabaie, J. Lin, and A. S. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," IEEE

Transactions on Automatic Control, vol. 48, no. 6, pp. 988–1001, June 2003.

[11] P. Barooah and J. P. Hespanha, "Graph effective resistances and distributed control: Spectral properties and applications," in Proc. Of the 45th IEEE Conference on Decision and Control, December 2006,pp. 3479–3485.